

FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFF	111111	111111	AAA	AAA
FFF	111111	111111	AAA	AAA
FFF	111111	111111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111111111	111111111	AAA	AAA
FFF	111111111	111111111	AAA	AAA
FFF	111111111	111111111	AAA	AAA

```
IIIIII  NN  NN  IIIII  FFFFFFFF  CCCCCC  BBBB
IIIIII  NN  NN  IIIII  FFFFFFFF  CCCCCC  BBBB
II      NN  NN  II      FF      CC      BB
II      NN  NN  II      FF      CC      BB
II      NNNN  NN  II      FF      CC      BB
II      NNNN  NN  II      FFFFFFFF  CC      BBBB
II      NN  NN  II      FFFFFFFF  CC      BBBB
II      NN  NN  II      FF      CC      BB
II      NN  NN  II      FF      CC      BB
II      NN  NN  IIIII  FF      CC      BB
IIIIII  NN  NN  IIIII  FF      CCCCCC  BBBB
IIIIII  NN  NN  IIIII  FF      CCCCCC  BBBB
```

```
....
....
....
....
```

```
LL      IIIII  SSSSSSS
LL      IIIII  SSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSS
LL      II     SSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LLLLLLLL  IIIII  SSSSSSS
LLLLLLLL  IIIII  SSSSSSS
```



```
1 0001 0 MODULE INIFCB (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1
33 0033 1 FACILITY: F11ACP Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 These routines create and initialize a file control block
38 0038 1 from the given file header.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1 STARLET operating system, including privileged system services
43 0043 1 and internal exec routines. These routines must be called in
44 0044 1 kernel mode.
45 0045 1
46 0046 1
47 0047 1 --
48 0048 1
49 0049 1
50 0050 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 14-Dec-1976 16:48
51 0051 1
52 0052 1 MODIFIED BY:
53 0053 1
54 0054 1 V03-002 LMP0221 L. Mark Pilant, 31-Mar-1984 12:16
55 0055 1 Add support for an ORB in the fcb.
56 0056 1
57 0057 1 V03-001 STJ3108 Steven T. Jeffreys, 24-Jun-1983
```

```

58      0058 1 | Fix link truncation error.
59      0059 1 |
60      0060 1 | A0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:01
61      0061 1 | Previous revision history moved to F11A.REV
62      0062 1 |
63      0063 1 | **
64      0064 1 |
65      0065 1 |
66      0066 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
67      0067 1 | REQUIRE 'SRCS:FCPDEF.B32';

```



```

69 0382 1 GLOBAL ROUTINE INIT_FCB (FCB, HEADER) : NOVALUE =
70 0383 1
71 0384 1 ++
72 0385 1
73 0386 1 FUNCTIONAL DESCRIPTION:
74 0387 1
75 0388 1 This routine initializes the given FCB according to the given
76 0389 1 file header.
77 0390 1
78 0391 1 CALLING SEQUENCE:
79 0392 1 INIT_FCB (ARG1, ARG2)
80 0393 1
81 0394 1 INPUT PARAMETERS:
82 0395 1 ARG1: FCB address
83 0396 1 ARG2: header address
84 0397 1
85 0398 1 IMPLICIT INPUTS:
86 0399 1 HEADER_LBN contains LBN of header block
87 0400 1
88 0401 1 OUTPUT PARAMETERS:
89 0402 1 NONE
90 0403 1
91 0404 1 IMPLICIT OUTPUTS:
92 0405 1 NONE
93 0406 1
94 0407 1 ROUTINE VALUE:
95 0408 1 NONE
96 0409 1
97 0410 1 SIDE EFFECTS:
98 0411 1 FCB initialized
99 0412 1
100 0413 1 --
101 0414 1
102 0415 2 BEGIN
103 0416 2
104 0417 2 MAP
105 0418 2 FCB : REF BBLOCK, ! FCB argument
106 0419 2 HEADER : REF BBLOCK, ! file header arg
107 0420 2
108 0421 2 LOCAL
109 0422 2 FCB_ORB : REF BBLOCK, ! Address of the ORB within the FCB
110 0423 2 MAP_AREA : REF BBLOCK, ! pointer to header map area
111 0424 2 MAP_COUNT : REF BBLOCK, ! count of map pointers
112 0425 2 MAP_POINTER : REF BBLOCK, ! pointer to scan map
113 0426 2 FILESIZE; ! size of file in blocks
114 0427 2
115 0428 2 EXTERNAL
116 0429 2 HEADER_LBN : ADDRESSING_MODE (GENERAL); ! LBN of file header
117 0430 2
118 0431 2 ! Set up the ORB address.
119 0432 2
120 0433 2 FCB_ORB = FCB[FCB$R_ORB];
121 0434 2
122 0435 2 ! Get the known constants and the simple stuff from the file header
123 0436 2 (i.e., header LBN, file ID, starting VBN, file owner and file protection).
124 0437 2
125 0438 2
```

```

126 0439 2 FCB[FCB$L_HDLBN] = .HEADER_LBN;
127 0440 2 FCB[FCB$W_FID_NUM] = .HEADER[FH1$W_FID_NUM];
128 0441 2 FCB[FCB$W_FID_SEQ] = .HEADER[FH1$W_FID_SEQ];
129 0442 2 FCB_ORB[ORB$W_UICMEMBER] = .HEADER[FH1$B_UICMEMBER];
130 0443 2 FCB_ORB[ORB$W_UICGROUP] = .HEADER[FH1$B_UICGROUP];
131 0444 2 FCB_ORB[ORB$V_PROT_16] = 1;
132 0445 2 FCB_ORB[ORB$W_PROT] = .HEADER[FH1$W_FILEPROT];
133 0446 2 IF .HEADER[FH1$V_SPOOL] THEN FCB[FCB$V_SPOOL] = 1;
134 0447 2 FCB[FCB$L_EFBLK] = ROT (.BBLOCK[HEADER[FH1$W_RECATTR], FAT$L_EFBLK], 16);
135 0448 2 IF .FCB[FCB$L_EFBLK] NEQ 0
136 0449 2 AND .BBLOCK[HEADER[FH1$W_RECATTR], FAT$W_FFBYTE] EQL 0
137 0450 2 THEN FCB[FCB$L_EFBLK] = .FCB[FCB$L_EFBLK] - 1;
138 0451 2
139 0452 2 ! Now scan the map area. Get the starting LBN if the file is contiguous
140 0453 2 ! and count up the file size from the retrieval pointers.
141 0454 2 !
142 0455 2
143 0456 2 MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
144 0457 2 MAP_POINTER = .MAP_AREA + FM1$C_POINTERS;
145 0458 2 FCB[FCB$W_SEGN] = .MAP_AREA[FM1$B_EX_SEGNUM];
146 0459 2
147 0460 2 FCB[FCB$L_STLBN] = 0; ! assume non-contiguous file
148 0461 2 IF .HEADER[FH1$V_CONTIG]
149 0462 2 THEN
150 0463 2 BEGIN
151 0464 2 FCB[FCB$L_STLBN] = .MAP_POINTER[FM1$W_LOWLBN]; ! get low order LBN
152 0465 2 (FCB[FCB$L_STLBN]<16,8) = .MAP_POINTER[FM1$B_HIGHLBN]; ! and high order
153 0466 2 END;
154 0467 2
155 0468 2 FILESIZE = 0;
156 0469 2 DECR MAP_COUNT FROM .MAP_AREA[FM1$B_INUSE]/2 TO 1 DO
157 0470 2 BEGIN
158 0471 2 FILESIZE = .FILESIZE + .MAP_POINTER[FM1$B_COUNT] + 1;
159 0472 2 MAP_POINTER = .MAP_POINTER + 4;
160 0473 2 END;
161 0474 2 FCB[FCB$L_FILESIZE] = .FILESIZE;
162 0475 2
163 0476 2 IF .FCB[FCB$L_EFBLK] GTR .FILESIZE
164 0477 2 THEN FCB[FCB$L_EFBLK] = .FILESIZE;
165 0478 2
166 0479 1 END; ! end of routine INIT_FCB
```

```

53 04 AC 003C 00000
50 58 A3 9E 00006
34 A3 00000006 00 D0 0000A
52 08 AC D0 00012
24 A3 02 A2 D0 00016
60 08 A2 9B 0001B
```

```

.TITLE INIFCB
.IDENT \V04-000\
.EXTRN HEADER_LBN
.PSECT $CODE$,NOWRT,2
.ENTRY INIT_FCB, Save R2,R3,R4,R5
MOVL FCB, R3
MOVAB 88(R3), FCB_ORB
MOVL HEADER_LBN, -52(R3)
MOVL HEADER, R2
MOVL 2(R2), 36(R3)
MOVZBW 8(R2), (FCB_ORB)
```

```

: 0382
: 0433
: 0439
: 0440
: 0442
```


	02	A0	09	A2	9B	0001F	MOVZBW	9(R2), 2(FCB_ORB)	:	0443
	0B	A0		01	88	00024	BISB2	#1, 11(FCB_ORB)	:	0444
	18	A0	0A	A2	B0	00028	MOVW	10(R2), 24(FCB_ORB)	:	0445
04	0D	A2		04	E1	0002D	BBC	#4, 13(R2), 1\$:	0446
	22	A3		10	88	00032	BISB2	#16, 34(R3)	:	
		55	3C	A3	9E	00036	MOVAB	60(R3), R5	:	0447
65	16	A2		10	9C	0003A	ROTL	#16, 22(R2), (R5)	:	
				07	13	0003F	BEQL	2\$:	0448
			1A	A2	B5	00041	TSW	26(R2)	:	0449
				02	12	00044	BNEI	2\$:	
				65	D7	00046	DECL	(R5)	:	0450
		50	01	A2	9A	00048	MOVZBL	1(R2), R0	:	0456
		51		62	40	3E	MOVAB	(R2)[R0], MAP_AREA	:	
		50	0A	A1	9E	00050	MOVAB	10(R1), MAP_POINTER	:	0457
	2A	A3		61	9B	00054	MOVZBW	(MAP_AREA), -42(R3)	:	0458
			30	A3	D4	00058	CLRL	8(R3)	:	0460
			0C	A2	95	0005B	TSTB	12(R2)	:	0461
				09	18	0005E	BGEQ	3\$:	
	30	A3	02	A0	3C	00060	MOVZWL	2(MAP_POINTER), 48(R3)	:	0464
	32	A3		60	90	00065	MOVB	(MAP_POINTER), 50(R3)	:	0465
				52	D4	00069	CLRL	FILESIZE	:	0468
		54	08	A1	9A	0006B	MOVZBL	8(MAP_AREA), R4	:	0469
		54		02	C6	0006F	DIVL2	#2, R4	:	
				54	D6	00072	INCL	MAP_COUNT	:	
		51		0C	11	00074	BRB	5\$:	
		52	01	A0	9A	00076	MOVZBL	1(MAP_POINTER), R1	:	0471
		50	01	A1	42	9E	MOVAB	1(R1)[FILESIZE], FILESIZE	:	
		F1		04	C0	0007F	ADDL2	#4, MAP_POINTER	:	0472
		A3		54	F5	00082	SOBGR	MAP_COUNT, 4\$:	0469
38		52		52	D0	00085	MOVL	FILESIZE, 56(R3)	:	0474
				65	D1	00089	CMPL	(R5), FILESIZE	:	0476
				03	15	0008C	BLEQ	6\$:	
		65		52	D0	0008E	MOVL	FILESIZE, (R5)	:	0477
				04	00	0091	RET		:	0479

; Routine Size: 146 bytes, Routine Base: \$CODE\$ + 0000

: 167 0480 1
: 168 0481 1 END
: 169 0482 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	146	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

INIFCB
V04-000

F 14
16-Sep-1984 01:07:36
14-Sep-1984 12:29:38

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]INIFCB.B32;1 Page 6
(2)

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
;\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	31	0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:INIFCB/OBJ=OBJ\$:INIFCB MSRC\$:INIFCB/UPDATE=(ENHS:INIFCB)

: Size: 146 code + 0 data bytes
: Run Time: 00:07.9
: Elapsed Time: 00:26.3
: Lines/CPU Min: 3642
: Lexemes/CPU-Min: 17410
: Memory Used: 110 pages
: Compilation Complete

0165 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

